CLAIMS

What is claimed is:

1. A method of maintaining the order of nodes in a hierarchical document, comprising:

5   selecting a first parameter corresponding to a selected maximum number of children for each node for an auxiliary ordered tree;

selecting a second parameter corresponding to a selected minimum number of children for each node for an auxiliary ordered tree;

building the auxiliary ordered tree having at least as many leaves as

10   atoms within said hierarchical document based upon the first and second parameters;

attaching the atoms to the leaves of said auxiliary ordered tree; and

labeling each of the nodes in the auxiliary ordered tree.

15   2. The method of claim 1, wherein the labeling of the nodes in the auxiliary tree is defined by:

$$N(\text{root}) = 0;$$
$$N(x) = N(y) + i \cdot (f - 1)^{h(x)}; \text{ and}$$
$$0 \le i < f$$

20   Where:

N(x) is the label for node x;

x is the ith child of y;

f is the maximum number of children per node; and

h(x) is the height of node x.

25

3. The method of claim 1, further comprising assigning labels to the atoms in the hierarchical document based upon the labels assigned to the corresponding leaves in the auxiliary ordered tree.

30   4. The method of claim 1, further comprising storing the labels of the

leaves of the auxiliary ordered tree.

5. The method of claim 4, further comprising storing the remaining portion of the auxiliary ordered tree.

5

6. The method of claim 1, further comprising partitioning the auxiliary ordered tree into a first portion that comprises the leaves from the auxiliary ordered tree and a second portion that comprises the remaining portion of the auxiliary ordered tree.

10

7. The method of claim 1, further comprising re-assigning labels to the atoms in the hierarchical document based upon the labels assigned to the corresponding leaves in the updated auxiliary ordered tree.

15 8. A method of updating an auxiliary ordered tree having at least as many leaves as atoms within a hierarchical document based upon a selected maximum number of children for each node and a selected minimum number of children for each node, comprising:

receiving a request to insert the hierarchical document with a new

20 atom at specific position;

inserting a new leaf in the auxiliary ordered tree based on the specific position of the corresponding atom in the hierarchical document;

searching for the highest ancestor node of the new leaf that has a number of leaves that equals or exceeds the selected maximum number of

25 leaves;

if no ancestor is found that equals or exceeds the selected maximum number of leaves then re-labeling the sub-tree rooted at the parent node of the new leaf;

if an ancestor node is found that has a number of leaves that equals or

30 exceeds the selected maximum number of leaves, then

determining whether the ancestor node is the root node;

if the ancestor node is the root node, then creating a new root having a predetermined number of children;

5

if the ancestor node is not the root node, then splitting the ancestor node into complete sub-trees that have the same leaf sequence as the ancestor node's sub-tree; and

reassigning labels in a top-down fashion in the sub-tree rooted at the parent of the ancestor node.

10 9. The method of claim 8, wherein the predetermined maximum number of leaves is defined as:

$$L_{max}(t) = s \cdot (f / s)^{h(t)}$$

Where:

f is a predetermined maximum fanout; and

15 s is a predetermined split factor.

10. The method of claim 8, wherein said insertion request comprises a request to insert a plurality of consecutive atoms and wherein said updating minimizes the cost of inserting the new leaves that correspond to the plurality

20 of consecutive atoms.

11. The method of claim 10, wherein the plurality of consecutive atoms comprise a plurality of tags and text segments.

25 12. The method of claim 8, further comprising:

receiving a request to delete an atom in the hierarchical document at a specific position; and

marking the corresponding leaf in the auxiliary ordered tree as deleted.

30 13. The method of claim 12, further comprising:

YOR920030239US1

determining whether the sub-tree of an ancestor node that equals or exceeds a predetermined maximum number of leaves has a sub-tree with a deleted leaf; and

inserting a new leaf in place of the deleted leaf.

5

14. A method of optimizing an auxiliary ordered tree having at least as many leaves as atoms within a hierarchical document, the shape of the auxiliary ordered tree being based upon a selected maximum number of children for each node and a selected minimum number of children for each

10 node, the method comprising adjusting the maximum number of children for each node and the selected minimum number of children for each node of the auxiliary ordered tree based upon application requirements regarding one of update cost, total cost of queries and updates, and the size of the labels.

15 15. A method of encoding an auxiliary ordered tree having at least as many leaves as atoms within a hierarchical document, the shape of the auxiliary ordered tree being based upon a selected maximum number of children for each node and a selected minimum number of children for each node, the method comprising minimizing space requirements using a virtual tree.

20

16. A system for maintaining the order of nodes in a hierarchical document, comprising:

means for selecting a first parameter corresponding to a selected maximum number of children for each node for an auxiliary ordered tree;

25 means for selecting a second parameter corresponding to a selected minimum number of children for each node for an auxiliary ordered tree;

means for building the auxiliary ordered tree having at least as many leaves as atoms within said hierarchical document based upon the first and second parameters;

30 means for attaching the atoms to the leaves of said auxiliary ordered

tree; and

    means for labeling each of the nodes in the auxiliary ordered tree.

17.   The system of claim 16, wherein the means for labeling each of the nodes bases the labeling upon:

$$N(\text{root}) = 0;$$
$$N(x) = N(y) + i \cdot (f - 1)^{h(x)}; \text{ and}$$
$$0 \leq i < f$$

Where:

N(x) is the label for node x;

x is the i$^{\text{th}}$ child of y;

f is the maximum number of children per node; and

h(x) is the height of node x.

18.   The system of claim 16, further comprising means for storing the labels of the leaves of the auxiliary ordered tree.

19.   The system of claim 18, further comprising means for storing the remaining portion of the auxiliary ordered tree.

20.   The system of claim 16, further comprising means for partitioning the auxiliary ordered tree into a first portion that comprises the leaves from the auxiliary ordered tree and a second portion that comprises the remaining portion of the auxiliary ordered tree.

21.   The system of claim 16, further comprising means for re-assigning labels to the atoms in the hierarchical document based upon the labels assigned to the corresponding leaves in the updated auxiliary ordered tree.

22.   A recording medium storing a program for making a computer

maintain the order of nodes in an hierarchical document, the program comprising:

instructions for selecting a first parameter corresponding to a selected maximum number of children for each node for an auxiliary ordered tree;

5   instructions for selecting a second parameter corresponding to a selected minimum number of children for each node for an auxiliary ordered tree;

instructions for building the auxiliary ordered tree having at least as many leaves as atoms within said hierarchical document based upon the first

10   and second parameters;

instructions for attaching the atoms to the leaves of said auxiliary ordered tree; and

instructions for labeling each of the nodes in the auxiliary ordered tree.

15   23.   The medium of claim 22, wherein the instructions for labeling each of the nodes is based upon:

$$N(root) = 0;$$
$$N(x) = N(y) + i \cdot (f - 1)^{h(x)}; \text{ and}$$
$$0 \leq i < f$$

20   Where:

N(x) is the label for node x;

x is the ith child of y;

f is the maximum number of children per node; and

h(x) is the height of node x.

25

24.   The medium of claim 22, further comprising instructions for assigning labels to the atoms in the hierarchical document based upon the labels assigned to the corresponding leaves in the auxiliary ordered tree.

30   25.   The medium of claim 22, further comprising instructions for storing

the labels of the leaves of the auxiliary ordered tree.

26.   The medium of claim 25, further comprising instructions for storing the remaining portion of the auxiliary ordered tree.

5

27.   The medium of claim 22, further comprising instructions for partitioning the auxiliary ordered tree into a first portion that comprises the leaves from the auxiliary ordered tree and a second portion that comprises the remaining portion of the auxiliary ordered tree.

10

28.   The medium of claim 22, further comprising instructions for re-assigning labels to the atoms in the hierarchical document based upon the labels assigned to the corresponding leaves in the updated auxiliary ordered tree.

15

29.   A system for updating an auxiliary ordered tree having at least as many leaves as atoms within a hierarchical document based upon a selected maximum number of children for each node and a selected minimum number of children for each node, comprising:

20   means for receiving a request to insert the hierarchical document with a new atom at specific position;

            means for inserting a new leaf in the auxiliary ordered tree based on the specific position of the corresponding atom in the hierarchical document;

            means for searching for the highest ancestor node of the new leaf that

25   has a number of leaves that equals or exceeds the selected maximum number of leaves;

            if no ancestor is found that equals or exceeds the selected maximum number of leaves then means for re-labelling the sub-tree rooted at the parent node of the new leaf;

30   if an ancestor node is found that has a number of leaves that equals or

exceeds the selected maximum number of leaves, then

means for determining whether the ancestor node is the root node;

if the ancestor node is the root node, then means for creating a new root having a predetermined number of children;

if the ancestor node is not the root node, then means for splitting the ancestor node into complete sub-trees that have the same leaf sequence as the ancestor node's sub-tree; and

means for reassigning labels in a top-down fashion in the sub-tree rooted at the parent of the ancestor node.

30.   A recording medium storing a program for making a computer update an auxiliary ordered tree having at least as many leaves as atoms within a hierarchical document based upon a selected maximum number of children for each node and a selected minimum number of children for each node, comprising:

instructions for receiving a request to insert the hierarchical document with a new atom at specific position;

instructions for inserting a new leaf in the auxiliary ordered tree based on the specific position of the corresponding atom in the hierarchical document;

instructions for searching for the highest ancestor node of the new leaf that has a number of leaves that equals or exceeds the selected maximum number of leaves;

if no ancestor is found that equals or exceeds the selected maximum number of leaves then instructions for re-labeling the sub-tree rooted at the parent node of the new leaf;

if an ancestor node is found that has a number of leaves that equals or exceeds the selected maximum number of leaves, then

instructions for determining whether the ancestor node is the

root node;

if the ancestor node is the root node, then instructions for creating a new root having a predetermined number of children;

if the ancestor node is not the root node, then instructions for splitting the ancestor node into complete sub-trees that have the same leaf sequence as the ancestor node's sub-tree; and

instructions for reassigning labels in a top-down fashion in the sub-tree rooted at the parent of the ancestor node.